Les architectures linguistiques et computationnelles en traduction automatique sont indépendantes

Christian BOITET

Laboratoire LIG, GETALP – Université Joseph Fourier,
385 rue de la bibliothèque, BP 53, 38041 Grenoble, Cedex 9, France
Christian.Boitet@imag.fr

Résumé Contrairement à une idée répandue, les architectures linguistiques et computationnelles des systèmes de traduction automatique sont indépendantes. Les premières concernent le choix des représentations intermédiaires, les secondes le type d'algorithme, de programmation et de ressources utilisés. Il est ainsi possible d'utiliser des méthodes de calcul « expertes » ou « empiriques » pour construire diverses phases ou modules de systèmes d'architectures linguistiques variées. Nous terminons en donnant quelques éléments pour le choix de ces architectures en fonction des situations traductionnelles et des ressources disponibles, en termes de dictionnaires, de corpus, et de compétences humaines.

Abstract Contrary to a wide-spread idea, the linguistic and computational architectures of MT systems are independent. The former concern the choice of the intermediate representations, the latter the type of algorithm, programming, and resources used. It is thus possible to use "expert" or "empirical" computational methods to build various phases or modules of systems having various linguistic architectures. We finish by giving some elements for choosing these architectures depending on the translational situations and the available resources, in terms of dictionaries, corpora, and human competences.

Mots-clés : Traduction Automatique, TA, TAO, architecture linguistique, architecture computationnelle, TA experte, TA par règles, TA empirique, TA statistique, TA par l'exemple

Keywords: Machine Translation, MT, linguistic architecture, computational architecture, expert MT, rule-based MT, empirical MT, statistical MT, example-based MT.

Introduction

Il y a un certain nombre d'idées fausses qui circulent parmi les chercheurs en TA, et freinent à notre avis les progrès dans ce domaine. La première est que la plupart des systèmes opérationnels utilisent la TA statistique, alors que la plupart (voir le Compendium (Hutchins & al. 2005) publié par l'EAMT) utilisent des méthodes « expertes » (« à règles », mais pas seulement à règles). L'autre est que les systèmes utilisant un « pivot interlingue », évidemment très adapté à la communication multilingue, sont nécessairement « à règles » (TAFR, en anglais RBMT ou « rule-based MT »), et donc très coûteux à construire (ce « donc » est faux aussi…).

Il ne faut pas faire l'amalgame entre l'architecture linguistique d'un système de TA, caractérisée par les représentations intermédiaires qu'il utilise durant le processus de traduction, et son architecture computationnelle, caractérisée par les méthodes de calcul et les ressources utilisées dans ses diverses « phases » transformant une représentation intermédiaire en sa suivante dans le processus.

Après une brève partie consacrée aux définitions des variantes de ces architectures, nous montrerons que, pour à peu près chaque architecture linguistique, on trouve des systèmes utilisant diverses architectures computationnelles. De plus, une bonne partie des systèmes utilisent plusieurs architectures computationnelles dans leurs différentes phases. Nous essaierons enfin de dégager quelques indications sur les choix d'architecture appropriés aux diverses situations traductionnelles et des ressources disponibles, en termes de dictionnaires, de corpus, et de compétences humaines.

1 Architectures des systèmes de TA

1.1 Architectures linguistiques

Ces architectures correspondent aux « chemins » dans le fameux « triangle de Vauquois ».

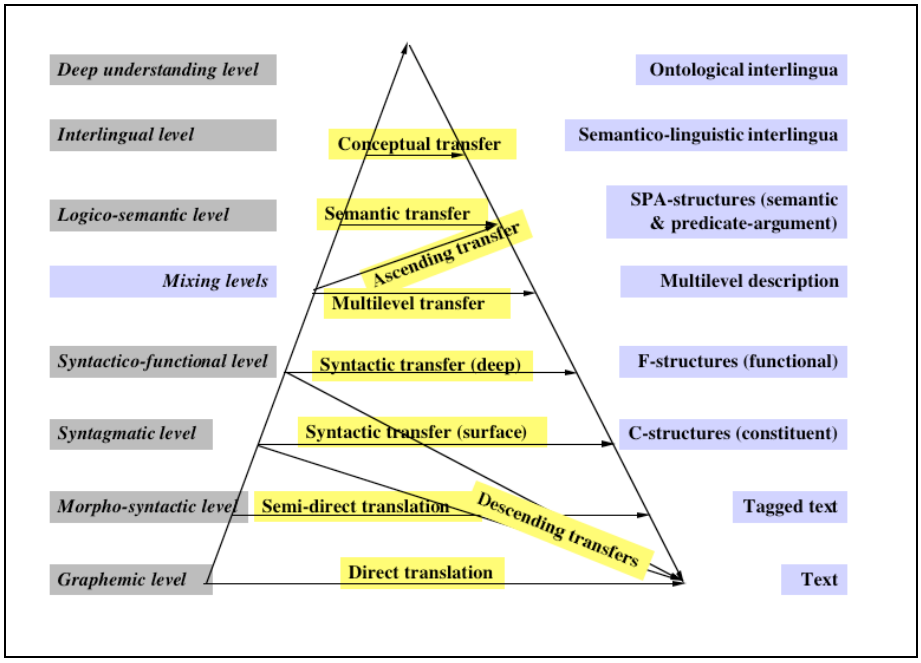

Figure 1 : triangle de Vauquois (Vauquois & Boitet 1985, Analectes de Vauquois — Boitet 1988)

Les systèmes *directs* n'utilisent que deux représentations, le texte d'entrée et le texte de sortie. Pour les langues ayant des systèmes d'écriture à séparateurs de mots ou de syllabes, le texte d'entrée n'est souvent pas strictement le flot de caractères tel quel, mais une suite de « mots typographiques » séparés grâce à des règles simples. Les systèmes *semi-directs* ont une phase de segmentation ou d'analyse morphologique, voire morphosyntaxique, et une phase de génération morphologique. C'est le cas des systèmes de « première génération » (russe-anglais aux USA et anglais-russe en URSS dès les années 1950), et un certain nombre de systèmes commerciaux actuels sont toujours de ce type.

Il existe au moins 7 variantes des systèmes à *transfert*. La structure obtenue en fin d'analyse peut être syntagmatique (basée sur des constituants la plupart du temps connexes), ou bien dépendancielle, et dans ce cas surfacique (fonctions syntaxiques comme sujet, objet direct, épithète, attribut…) ou profonde (relations sémantiques comme agent, patient, cause, concession…). Les systèmes à *transfert profond* fondés sur les théories de Tesnière, puis de l'Ecole de Prague et de celle de Moscou, utilisent des représentations logico-sémantiques distinguant les arguments des circonstants.[1]

[1] Les circonstants portent des relations sémantiques (cas profonds), tandis que les arguments ne portent en général qu'un numéro (Arg0, Arg1… Arg4 ou Arg5 au maximum), car il est très difficile sinon impossible d'affecter fiablement une relation sémantique à un argument, si le répertoire de ces relations est celui utilisé pour les circonstants. Le projet FrameNet montre d'ailleurs bien que, si on veut définir des relations

On dit qu'il y a « transfert lexical » quand on passe directement de « l'espace lexical » de la langue source à celui de la langue cible. Par « espace lexical », on entend tout le système lexical, qui va des « formes » de surface aux « acceptions », en passant par les lemmes et éventuellement par les « unités lexicales » (familles dérivationnelles) ou « prolexèmes » (les mêmes, un peu élargies).

Le « pivot hybride » (terme dû à Shaumyan) des systèmes du CETA des années 1965-70 était un type de représentation utilisant des attributs et relations interlingues, et des unités lexicales de chacune des langues. Ces systèmes étaient donc à transfert simple, alors qu'on a un double transfert en « pivot ».

Les structures *multiniveau* de Vauquois sont basées sur un graphe syntagmatique abstrait (suppression des auxiliaires, regroupement de lexèmes discontinus comme give...up, etc.), lexicalisé (chaque nœud interne domine un « gouverneur » lexical), et contenant aussi bien les informations et relations profondes que celles de surface. De telles structures sont « génératrices » des structures mononiveau usuelles, et offrent une sorte de « filet de sécurité ».

Les systèmes à véritable *interlingua* (comme ATLAS-II de Fujitsu ou PIVOT/Crossroads de NEC, ou KANT/CATALYST de CMU/Caterpillar, ou UNL, ou MASTOR-1 d'IBM, ce dernier en TA de parole) utilisent 3 espaces lexicaux, car un véritable interlingua possède son propre vocabulaire, même si ce vocabulaire est construit comme union des acceptions[2] d'un certain nombre de langues, comme UNL (Uchida 1996, 2004). Dans les systèmes de TA, il existe des interlinguas « linguistico-sémantiques » (comme KANT, ULTRA, UNL) dont les « lexèmes » sont construits à partir des lemmes et des lexies de dictionnaires d'une ou plusieurs langues naturelles, et des interlinguas « sémantiques » ou « sémantico-pragmatiques », dont les lexèmes sont construits à partir des entités, propriétés, actions et processus d'un domaine précis et d'un ensemble de tâches bien identifiées (par exemple, réservation touristique).

Enfin, si la plupart des systèmes de TA ont comme « unité de traduction » le « segment » (phrase ou titre) des systèmes d'aide au traducteur utilisant des mémoires de traductions, certains ont des unités de traduction de l'ordre de la page (Ariane-G5), ce qui permet de mieux traiter certains phénomènes comme la concordance des temps et de résoudre des anaphores hors du contexte de la phrase.

1.2 Architectures computationnelles

Pour ce qui est des processus automatiques, on distingue entre méthodes *expertes* et méthodes *empiriques*. Il y a aussi des distinctions à faire si le processus de traduction est interactif.

1.2.1 Méthodes « expertes »

Les méthodes « expertes » sont plus ou moins procédurales ou déclaratives, et font appel à de la programmation directe ou fondée sur des « modèles de calcul » abstraits, d'où l'utilisation de *LSPL* (langages spécialisés pour a programmation linguistique). On a en bref :

- la programmation directe dans un langage algorithmique classique (souvent employée au niveau des traitements typographiques ou morphologiques).
- la programmation directe dans un langage de haut ou très haut niveau (Lisp, Prolog) offrant des structures de données et de contrôle plus adaptées à la programmation linguistique, mais demandant une grande expertise en programmation.
- la programmation dans des LSPL d'automates (comme les transducteurs finis, les ATN ou les transformateurs d'arbres, abusivement dits « grammaires » transformationnelles).
- la programmation dans des formalismes de grammaires déclaratives (ou presque) comme LFG, GPSG, HPSG, ou TAG.

---

sémantiques pour les arguments, il faut le plus souvent les lexicaliser ("donner" aura alors "donateur/donneur" pour Arg0, "don" pour Arg1, "donataire" pour Arg2).

[2] Une "acception" est un sens d'un mot, au sens de lemme ou terme, dans l'usage de la langue. Une "lexie" est un sens de mot dans un dictionnaire.

Il est abusif de parler de systèmes « à règles » pour les deux premiers. Ainsi, Systran utilise des automates (transducteurs d'états finis) pour l'analyse morphologique, tandis que l'analyse syntaxique n'est pas faite par « règles », mais par un programme instanciant un schéma procédural fixe (écriture de « macros » déterminant des décisions locales par examen d'une « fenêtre courante » sur un graphe sans boucle représentant la phrase).

1.2.2 Méthodes empiriques

Ce sont les méthodes fondées sur les corpus :

- TA statistique (SMT) et TA statistique à syntagmes (PSMT, ou « phrase-based » SMT),
- TA fondée sur les exemples (EBMT), avec 3 variantes.

Notons que « TA statistique » est un assez mauvais terme, car on devrait plutôt parler de TA « probabiliste ». En effet, un « modèle de langage » est une collection de probabilités estimées d'après des comptages sur de gros ou très gros corpus.

La différence essentielle entre SMT et EBMT est que, en EBMT, les exemples sont utilisés directement durant le processus de traduction, tandis que la SMT utilise les résultats d'une sorte de gigantesque « compilation » de l'ensemble des exemples (corpus aligné).

Les variantes de l'EBMT sont les suivantes :

- En *EBMT classique,* on étend les techniques de recherche de segments voisins des systèmes d'aide aux traducteurs avec mémoire de traductions, et on propose, pour les mots différenciant le segment à traduire et le segment trouvé, des remplacements venant d'autres exemples ou de dictionnaires. Le système Similis™ (dérivé de (Planas 1998)) d'aide au traducteur en est proche.
- En *EBMT par analogie* (Lepage & Denoual 2005), si S1 est le segment à traduire (en langue L1), on cherche les « rectangles analogiques » `P1:Q1::R1:S1` tels qu'on dispose des exemples de traduction `(P1,P2), (Q1,Q2), (R1,R2),` et on résout en `X` (dans la langue L2) l'équation analogique `P2:Q2::R2:X`. On obtient en général plusieurs traductions `X`, qu'on filtre pour la fluidité par un modèle n-gramme. Si on ne trouve pas de tel rectangle, on résout en `Y` (dans la langue L1) l'équation `P1:Q1::Y:S1` et on continue récursivement. Il n'y a donc pas de « décomposition en morceaux qui se correspondent » puis de « recomposition ».
- Dans le système *EBMT par exemples de correspondances structurées* de Al-Adhaileh et Tang (USM, Penang), on utilise un corpus parallèle annoté par des S-SSTC (correspondances chaîne-arbre structurées synchronisées). La traduction se fait par analyse-synthèse. Une correspondance `(C1, A1)—c—(C2, A2)` est élémentaire ou composée (= $\{(C1_i, A1_i)—c_i—(C2_i, A2_i)\}_i$). Quand on en trouve une car on a identifié un morceau C1 du segment S1 à traduire, ou bien les correspondances la constituant, on a d'un seul coup les 3 autres éléments et leur synchronisation.

2 Variété des architectures computationnelles

Voici maintenant une étude synthétique (non exhaustive) des architectures computationnelles utilisées dans des systèmes de TA basés sur 11 architectures linguistiques différentes. Pour la clarté, nous utilisons des tableaux, organisés de la façon la plus homogène possible. Il n'a malheureusement pas été possible de suivre la suggestion d'un relecteur, et de faire un seul grand tableau croisant les deux architectures, car trop de systèmes utilisent différentes architectures computationnelles dans différentes phases du traitement. Pour des raisons de place, il n'a pas non plus été possible de mettre autant de références qu'on l'aurait souhaité. D'un autre côté, les références sur les systèmes opérationnels (commerciaux comme Systran, ATLAS, The Translator, Honyaku-no-oo-sama, ProMT, Softissimo, Tracy, PIVOT/Crossroads, ALTFlash, METAL/Compendium, LanguageWeaver, etc., et non commerciaux ou semi-commerciaux comme PAHO-MTS, ALT/JE ou Google Translator) sont très rares et souvent anciennes. Le « Compendium » (Hutchins & al. 2005) est une source importante, mais ne donne pas de détails précis sur la façon dont les systèmes cités sont construits.

2.1 Systèmes de traduction directe

Type	Étapes	Méthode	Commentaires	Exemples
RBMT 1975—	Segmentation Trad. mot à mot	FST (règles + dict.) **règles**	Convient pour des langues très voisines japonais ↔ coréen, hindi ↔ urdu …	ATLAS-I Fujitsu,76-78 (coréen↔japonais)
SMT 1980—	Segmentation, réarrangement…	Alignement + « décodage » **statistique**	SMT = première idée sur la TA par les cryptographes de la 2° guerre mondiale (W. Weaver 1949)	Beaucoup de systèmes statistiques (SMT) IBM 1980-
EBMT 2000—	Pas de prétraitement EBMT « pure »	Résolution analogique + filtrage n-grammes **analogique**	Résultats ≈ ceux de la SMT Nagao 1984 (plutôt TA par similarité) Lepage 2000 (vraie analogie à 4 termes)	ALEPH ATR 2000- GREYT, Caen 2006—

Le plus souvent, ces systèmes sont « empiriques », mais certains utilisent une approche « experte », comme ATLAS-I (différent de ATLAS-II).

2.2 Systèmes de traduction semi-directe

Type	Étapes	Méthode	Commentaires	Exemples
1G-MT 1950—	Segmentation & Lemmatisation par programme	Consultation de dictionnaire + "macros" de réarrangement **procédural**	Tables + macros sur des chaînes **procédural**	GAT (Georgetown) EURATOM, Ispra, 1965—69 SPANAM-1, PAHO, ≈1975— GLOBALINK ← Spanam-1 (PAHO)
SMT 1990—	Lemmatisation Décodage	**Procédural** + règles **statistique**	Modèle de langue **probabiliste**	Candide IBM, 1980—, Google 2005- Beaucoup de systèmes statistiques
Trad. **pidgin**	Lemmatisation Transfert lexical Réarrangement Génération	Traitement de chaînes **règles en** systèmes-Q -- --	**procédural** (snobol4) Énoncé = graphe de chaînes d'arbres étiquetés	Idée de B. Harris (TAUM, « traductologiste ») rus → eng, fre (Boitet 1972)

GlobaLink a été fait à partir d'une copie de Spanam-1. Spanam-2 est de type expert (ATN).
Les systèmes actuels de Google sont (sans doute) plus PSMT (phrase-based SMT) que SMT.

2.3 Systèmes à transfert descendant de constituants

Type	Étapes	Méthode	Commentaires	Exemples
RBMT 1970—	Analyse par **ATN** Transfert/génération	LSPL étendant Lisp ou un autre Lprog Descente récursive	règles + dict. + transformation **procédural + règles**	ENGSPAN, SPANAM-2, ou 'PAHO-MTS' (PAHO, ≈1978—) AS-Transac (Toshiba, 1982—) Reverso ProMT, 1986—
RBMT 1980—	Analyse par **ECFG** (hors-contexte étendu) Transfert /génération	LSPL étendant Lisp ou un autre Lprog Descente récursive	grammaire+dict. **règles** **procédural + règles**	METAL (TUA+Siemens, 1982—) Duet-2 (Sharp, 1984—) Shalt-1 (IBM-Japon, 1982—) Kate KDD (1983—)
RBMT 1984—	Lemmatisation Slot-grammars T+G en Prolog	Dictionnaire +tables LSPL étendant Prolog Descente récursive	**procédural** **règles** **procédural + règles**	LMT (IBM-US, 1983—) PT-1 (Personal Tanslator) de Linguatec, dérivé de LMT, —2000

Les systèmes récents de type PSMT de LanguageWeaver sont sans doute aussi de ce type.

2.4 Systèmes à transfert descendant de dépendances

Type	Étapes	Méthode	Commentaires	Exemples
1.5G-MT 1990—	Lemmatisation Analyse produisant un graphe de dépendances Transfert /génération	FST (+ dictionnaires) macros C + dictionnaire Descente récursive	**règles** **procédural** **procédural**	Systran 1990—
RBMT 1985—	Segm.+ lemmatisation Analyse de dépendances Désamb. interactive Transfert /génération	dictionnaire + tables LSPL pour les grammaires de dépendances + contraintes limitées (1 seul quantificateur) Descente récursive	**procédural** grammaire + dict. **règles** **contraintes** **procédural + règles**	JETS (IBM-Japon, 1985-90)

RBMT /SMT 2000—	Segmentation et lemmatisation multiple Analyse de dépendances Tranfert/génération	**Programmation** en Pascal puis C + dictionnaire Algorithme factorisant (DP) Descente récursive	**procédural + règles** Version hybride (TA experte + SMT) depuis 2006 **+ statistiques**	Neon (Xiamen) En-Ch & Ch-En, 2000—

Systran est très ancien (1966), mais depuis 1990 environ il intègre des FST pour les traitements morphologiques, et les macros utilisées pour la suite du traitement sont développées en C et plus en assembleur. Dans JETS (ancêtre de Honyaku no oo-sama, actuellement commercialisé par IBM-Japon), les dépendances sont les « cas profonds » correspondant aux particules casuelles du japonais.

2.5 Systèmes à transfert horizontal de constituants

Type	Analyse/données	Transfert/préparation	Génération/méthode	Exemples
RBMT 1995—	Lemmatisation + Slot Grammars **règles**	Contient la génération Prolog **procédural + dict.**	Descente récursive grammaire+dict. **règles**	PT (= LMT d'IBM) Linguatech, 1995—2000
EBMT 2000—	Données initiales: corpus // bilingue dictionnaire	Préparation: construction autom. de S-SSTCs puis édition (humaine)	Traduction: 3 étapes en parallèle (A//T//G) combinaison ascendante	« EBMT » (ou 'Banturjah') UTMK, USM, 2000— basé sur un corpus de S-SSTC
PSMT PSCFG 2002—	Lemmatisation Chunking **statistique**	Alignement Décodage **statistique**	Aplatissement de l'arbre Post-traitement **statistique**	LanguageWeaver 2002— Google 2005— +Wu, Melamed 1997, 2004

La différence avec les systèmes précédents est que le transfert produit une structure de même nature que ce que produirait l'analyse de l'unité de traduction cible. Cela permet éventuellement de composer deux systèmes de TA en perdant beaucoup moins d'information et en introduisant beaucoup moins d'erreurs qu'en mettant bout à bout deux systèmes complets, i.e. en passant par un « pivot textuel ».

2.6 Systèmes à transfert horizontal de dépendances

Type	Analyse	Transfert	Génération	Exemples
RBMT 1975—	Grammaire + dict. Anal. de dépendance **règles**	Dictionnaires Transformations d'arbres **règles**	Aplatissement de l'arbre grammaire+dict. **règles**	ETAP-2, ETAP-3 IPPI, Moscou, 1977—
RBMT 1992—	Lemmatisation + patrons linéaires **règles**	Dictionnaire de « treelets » + thesaurus sémantique **règles**	Aplatissement de l'arbre grammaire+dict. **règles**	TDMT, Furuse (prototype pour la TA de parole) ATR, 1992—1998
RBMT+ SMT 1999—	Analyseurs de MSR (Microsoft) **règles** (en G)	Apprentissage du transfert à partir de paires (lf_s, lf_t) **statistique**	Générateurs de Microsoft **règles** (en G)	MTS-1 (prototype sur de la documentation technique)

LMT (MacCord, IBM), est rangé ici car les « slots » correspondent à des fonctions syntaxiques.

2.7 Systèmes à transfert multiniveau horizontal

Type	Étapes	Méthode	Commentaires	Exemples
RBMT 1990—	Lemmatisation Analyse multiple par ECFG (gouvernement & liage) Désambiguïsation interactive si pas assez de place Transfert autonome Génération	Dictionnaire + tables Programmation en langage évolué (Modula) -- -- descente récursive	**procédural + dict.** dict. + grammaire **procédural + règles** **interactif** **procédural + règles** **+ dictionnaire** aplatissement de l'arbre	ITS-2 (Genève, 1990—)
RBMT 2000—	Lemmatisation + Slot Grammars Transfert autonome Génération	LSPL grammatical, analyse multiple dictionnaire de treelets descente récursive	**procédural + dict.** dict. + gram. **procédural + règles** (en Prolog)	PT-2 (Linguatec et Lingenio) depuis 2000

Passer d'une architecture à transfert descendant à celle de transfert « horizontal » a été très difficile (communication personnelle de K. Eberle de Linguatec à COLING-2000). Cela a été aussi tenté sur METAL (par Siemens puis Sietech), mais sans succès.

2.8 Systèmes à transfert ascendant multiniveau

Type	Étapes	Méthode	Commentaires	Exemples
RBMT 1978—	Analyse morphologique Analyse structurale Tansfert lexical Transfert structural Génération structurale Génération morphol.	dictionnaire + automate transformations d'arbres **règles de réécriture** dictionnaires transformations d'arbres dictionnaire + automate	LSPL (5 au total) **règles** pour toutes les phases **dictionnaires** pour certaines phases	Systèmes en Ariane-G5 1974- ru-de→ru, en→my-th 80-87 fr→en (BV/aero) 85-92 fr→en-de-ru (LIDIA) 90-96 HICATS Hitachi (1990-) Jemah USM, NUS (1990-)

Ici, le transfert produit une structure multiniveau « génératrice » dans laquelle les informations non interlingues correspondent à celles de la langue source de façon « contrastive », et sont à utiliser par le générateur comme des préférences ou des ordres en fonction des valeurs de certains attributs « tactiques ». La première phase de l'étape de génération consiste alors à « sélectionner une paraphrase » en recalculant les informations de surface.

2.9 Systèmes à transfert sémantique ou « conceptuel »

Type	Étapes	Méthode	Commentaires	Exemples
RBMT 1982—	Segm. +lemmatisation Autres phases	Programmation en C Transformations d'arbres	**procédural** **règle**s (gram.s + dict)	MU (Kyodai, 82-87) MAJESTIC (JICST, 87—)

On pourrait ajouter les systèmes du CETA (1962-70), à « pivot hybride », décrit plus haut.

2.10 Systèmes à interlingua sémantique ou « linguistico-sémantique »

Il s'agit de systèmes utilisant un interlingua muni d'un vocabulaire « autonome ».

Type	Enconversion	Déconversion	Commentaires	Exemples
RBMT 1980—	Lemmatisation directe Transformations chaîne-graphe **règles**	Transformations graphe-chaîne **règles**	**procédural** + **règles**	ATLAS-II Fujitsu, 1980— PIVOT Nec, 1983—
RBMT 1980—	Formalisme proche des DCG **règles**	LSPL fondé sur des **règles Prolog**	**règles**	ULTRA NMSU, 89-95
RBMT 1997—	Selon les partenaires **règles** (jusqu'à présent)	Selon les partenaires **règles**	graphe UNL = structure « anglo-sémantique »	UNL 1996—

Les graphes UNL sont « linguistico-sémantiques ». Le vocabulaire (UW) est l'union des acceptions des différentes langues traitées, comme dans ULTRA, mais les relations sémantiques et le traitement des idiomes sont liés à l'anglais (et tant mieux, car les langues voient assez souvent différemment les relations sémantiques dans des énoncés synonymes).

2.11 Systèmes à ontologie

Ces systèmes sont les seuls à faire de la « compréhension explicite », leur interlingua étant « projeté » dans une ontologie Ω, soit de façon séparée, soit de façon interne.

Type	Enconversion	Projection dans une Ω	Déconversion	Examples
KBMT 1980—	Lemmatisation & EPSG+f-structures +pseudo-unification **Règles** (Univ. Parser)	Oui, de tout sauf les éléments de discours dict. + **règles** + désamb. interactive	Planification de la structure profonde Descente récursive **règles**	**KBMT-89** CMU, 1989—91 **KANT/Catalyst** CMU+Caterpillar, en→fr-sp-de-? 1992—
RBMT 1997—	Dictionnaire + FST **règles**	Pas d'ontologie explicite séparée : c'est l'idée (ancienne) des « grammaires sémantiques »	dictionnaire + FST **règles**	CSTAR-II & Nespole! GETA 97-03, ETRI (Corée) 97-99
SMT 2003—	Appris à partir de couples (chaîne,IF) **statistique**		Appris à partir de couples (IF,chaîne) **statistique**	CSTAR-II & Nespole! Irst 98-03 Mastor-1 (IBM 2003), sur PDA

L'IF (interface format) réfère à une ontologie implicite, pas explicite.

3 Éléments pour le choix d'architectures en TA

3.1 Taille et coût des ressources / architectures computationnelles

Le tableau suivant donne une estimation des ressources nécessaires pour construire un système de TA en fonction de la difficulté de la tâche, grossièrement estimée à partir de la taille moyenne des phrases. Les coûts sont donnés ici en homme*année (h*a), M veut dire « million », et K « mille ».

- Pour la TA empirique, il s'agit de la taille du corpus, en mots, pages (de 250 mots), phrases, et du temps humain de préparation de ce corpus. S'il s'agit de traduction, nous utilisons le taux professionnel de 1h/page (avec la révision, ce serait 1h20 par page). S'il s'agit d'annotation, les coûts ne sont la plupart du temps pas publiés, et nous utilisons des informations dont nous disposons par communications personnelles. Le coût par page est bien plus élevé, mais le corpus peut être beaucoup plus petit, et finalement bien moins coûteux, pour de meilleurs résultats.
- Pour la TA experte, il s'agit de la taille des dictionnaires et des grammaires, et du travail d'experts humains. Contrairement à ce qu'on lit dans de nombreux cours sur la TA qu'on peut glaner sur le Web, ce coût est souvent très surévalué, et pas seulement par les tenants des méthodes empiriques.

Phrases / Type	6.5 mots/phrase BTEC, METEO	25 mots/phrase Informations (news)
SMT **PSMT** **EBMT par** analogie Coût :	0.9—3 M mots 3.6—12 K pages 0.15—0.5 M phrases **2.4—8 h*a**	50—200 M mots 200—800 K pages 2—8 M phrases **100—400 h*a** (rarement disponible !)
EBMT avec arbres **SMT** Mastor-1 (IBM) Coût :	N/A pour ce type de phrases courtes Apprentissage supervisé 1h/page (par recoupements)	4—12.5 M mots 15—50 K pages 0.15—0.5 M phrases **10—40 h*a**
EBMT avec arbres et S-SSTCs Banturjah (USM) Coût :	N/A pour phrases courtes Apprentissage supervisé 15 h/page (10 h/p espéré) dictionnaire (50 K) souvent disponible	4—12.5 M mots 0.6—1 K pages 0.006—0.01 M phrases **6—10 h*a** (travail assez spécialisé)
RBMT Coût :	Dictionnaire 3-10 K 0.6—2 h*a Total **1—3 h*a**	Dict. 50-500 K, soit 15—150 h*a Grammaires environ 25 h*a Total ≈ **40—175 h*a**

3.2 Brève analyse

1. Il est clair que, plus les corpus sont « bruts », plus ils doivent être grands. Même à raison de 15h/page de travail humain, il semble intéressant d'utiliser une méthode comme celle de l'USM à Penang, car on n'a besoin que de 1000 pages et d'un gros dictionnaire assez simple.
2. D'autre part, la SMT (et la PSMT) sont en fait adaptées à des « niches de riches », tout comme la TA « experte » pour sous-langages. En effet, il y a très peu de corpus parallèles disponibles de 200 à 800 K pages ! Du point de vue des corpus, les différences entre couples de langues « bien dotés » et « mal dotés » sont encore plus grandes qu'en ce qui concerne les dictionnaires.
3. Créer de très gros corpus parallèles à partir de zéro est 2 à 3 fois plus coûteux que de construire un grand système de TA par approche experte (procédurale et/ou à automates et grammaires).
4. L'architecture linguistique par « pivot interlingue » peut utiliser n'importe quel paradigme computationnel, qu'il soit statistique, analogique, à règles, ou hybride.
5. En dernier ressort, le choix de l'architecture linguistique et de l'architecture computationnelle dépend des ressources disponibles en termes de corpus préalablement traduits, et d'humains plus ou moins experts. Les types d'expertise recherchée sont, par ordre de difficulté croissante (estimée via le temps de formation et la relative rareté des experts) : la traduction, la post-édition, la correction d'annotations, l'annotation à partir de rien, la terminologie, la lexicographie complexe

(vocabulaire général et tournures), l'écriture de grammaires assez déclaratives, la programmation par automates dans des LSPL adaptés, et enfin la programmation directe.

Conclusion

Nous avons donc montré que les architectures linguistiques et computationnelles des systèmes de traduction automatique sont indépendantes, au sens où on peut utiliser n'importe quelle architecture computationnelle pour réaliser n'importe quelle phase de traitement dans une architecture linguistique donnée, non seulement en théorie, mais en pratique, comme l'illustre la variété des systèmes cités en exemple. Nous avons aussi donné une évaluation des tailles et des coûts de construction des ressources utilisées par différents types de systèmes de TA, ce qui donne quelques éléments pour le choix de l'architecture linguistique et computationnelle d'un système à créer, en fonction des situations traductionnelles et des ressources disponibles, en termes de dictionnaires, de corpus, et de compétences humaines.

Cette réflexion ouvre sur une perspective plus générale et « sociétale ». Si l'on veut surmonter la « barrière linguistique » entre toutes les langues, on ne pourra pas se contenter de construire des systèmes de TA entre l'anglais et les autres langues, même pas pour le tchat entre deux langues différentes de l'anglais. En effet, l'anglais intermédiaire serait nécessairement trop « grossier », entaché d'erreurs, et porteur d'ambiguïtés nouvelles en sus des anciennes (celles de la langue source). La plupart des locuteurs (ou simplement lecteurs « passifs ») seront de plus toujours bien moins compétents et à l'aise en anglais que dans leur langue.

Il faudra donc construire des systèmes fondés sur des interlingues, soit « sémantico-pragmatiques » (comme l'IF de CSTAR, Nespole! ou MASTOR-1) s'il s'agit de tâches et de domaines restreints et bien identifiés, soit « linguistico-sémantiques » (comme UNL). Cela sera d'autant plus nécessaire qu'on voudra intégrer ces systèmes au « Web sémantique », car il faudra alors demander aux internautes d'aider les systèmes d'annotation, sans doute par le même type de « désambiguïsation interactive » que celui qui permet de compenser la nécessaire « rusticité » (ou la « mauvaise qualité intrinsèque ») des systèmes de TA « tout terrain » quand on veut les utiliser en « tout automatique ».

Il ressort de ce qui précède qu'il devrait être possible de construire des systèmes de TA entre toutes les langues, passant par un niveau sémantique comme UNL, non seulement par des approches « expertes » comme c'est le cas actuellement, mais par des approches empiriques moins coûteuses et moins longues en développement, si toutefois on disposait de corpus adéquats de taille suffisante. D'autre part, à la lumière des développements récents en alignement et en TA statistique, de tels corpus devraient pouvoir être construits par « transitivité », en alignant des corpus parallèles et des corpus annotés en IL (en UNL par exemple) s'ils ont au moins une langue en commun.

Références

BOITET C. (1986) The French National MT-Project: technical organization and translation results of CALLIOPE-AERO. *Computers and Translation*, 1, pp. 281—309.

BOITET C. (1988) L'apport de Bernard Vauquois à la traduction automatique et au traitement automatique des langues naturelles. *Proc. Colloque sur l'Histoire de l'Informatique en France.,* 3-5 mai 1988, P. Châtelin, ed., vol. 2/2, pp. 63—82.

BOITET C. (1988) PROs and CONs of the pivot and transfer approaches in multilingual Machine Translation. *Proc. Int. Conf. on « New directions in Machine Translation »,* 18–19 August 1988, BSO, ed., Foris Publications, pp. 93—108.

BOITET C. (1988) Representation and Computation of Units of Translation for Machine Interpretation of Spoken Texts. *Computers and Artificial Intelligence*, **8**/6, pp. 505—546.

BOITET C. (1993) La TAO comme technologie scientifique : le cas de la TA fondée sur le dialogue. In *La traductique*, A. Clas et P. Bouillon, ed., Presses de l'Université de Montréal, pp. 109—148.

BOITET C. (1993) TA et TAO à Grenoble… 32 ans déjà ! *T.A.L. (revue semestrielle de l'ATALA)*, **33**/1—2, Spécial Trentenaire, pp. 45—84.

BOITET C. (1995) Factors for success (and failure) in Machine Translation — some lessons of the first 50 years of R&D. *Proc. MTS-V (Fifth Machine Translation Summit),* 11—13 July 1995, CEE, 17 p.

BOITET C. (2001) Machine Translation. In *Encyclopedia of Cognitive Science*, Nature Publishing Group, London, (in manuscript form) 24 p.

BOITET C. ET BLANCHON H. (1994) Promesses et problèmes de la « TAO pour tous » après LIDIA-1, une première maquette. *Langages*, 116, pp. 20—47.

BOITET C., BOGUSLAVSKIJ I. ET CARDEÑOSA I. (2007) An Evaluation of UNL Usability for High Quality Multilingualization and Projections for a Future UNL++ Language. In *Computational Linguistics and Intelligent Text Processing (Proc. CICLING-2007)*, A. Gelbukh, ed., Springer (LNCS 4394), pp. 361-373. (ISBN-10: 3-540-70938-X Springer, ISSN: 0302-9743)

BOITET C., ed. (1988) BERNARD VAUQUOIS et la TAO, vingt-cinq ans de Traduction Automatique, ANALECTES. BERNARD VAUQUOIS and MT, twenty-five years of MT. Ass. Champollion & GETA, Grenoble, 700 p.

BOITET C. ET GERBER R. (1986) Expert Systems and other new techniques in MT. In *Neue Ansätze in maschineller Sprachübersetzung*, Niemeyer, Tübingen, pp. 103—119.

EISELE A. (2005) *Exploiting Multilingual Corpora for Machine Translation*. (JRC Enlargement and Integration Workshop on Exploiting parallel corpora in up to 20 languages), Arona, Saarland University & DFKI, (slides)

HUTCHINS W. J. (1986) Machine Translation : Past, Present, Future. Ellis Horwood, John Wiley & Sons, Chichester, England, 382 p.

HUTCHINS W. J. ET SOMERS H. L. (1992) An Introduction to Machine Translation. H. B. Jovanovich, ed., Academic Press, 362 p.

HUTCHINS J., HARTMAN W. ET HITO E. (2005) Compendium of Translation Software (directory of machine translation systems and computer-aided translation support tools. EAMT, TIM/ISSCO, Geneva, 127 p. (Earlier editions of the Compendium are available as PDF files from: http://ourworld.compuserve.com/homepages/WJHutchins/compendium.htm)

JEIDA (1989) A Japanese view of Machine Translation in light of the considerations and recommendations reported by ALPAC, USA. Japanese Electronic Industry Development Association.

KRAIF O. (2006) Corpus multilingues — multilingual corpora. 22/11/06.
http://w3.u-grenoble3.fr/kraif/index.php?option=com_content&task=view&id=20&Itemid=36

LEPAGE Y. ET DENOUAL E. (2005) Purest ever example-based machine translation: detailed presentation and assessment. *Machine Translation Journal*, 19, pp. 251—282.

SÉNELLART J., BOITET C. ET ROMARY L. (2003) XML Machine Translation. *Proc. MTS-IX (Machine Translation Summit),* New-Orleans, 9 p.

THURMAIR G. (2006) Using corpus information to improve MT quality. *Proc. LR4Trans-III (3rd International Workshop on Language Resources for Translation Work, Research & Training), LREC 2006,* Genoa, ELRA / ELDA, 4 p.

UCHIDA H. (2004) The Universal Networking Language (UNL) Specifications Version 3 Edition 3. UNL Center, UNDL Foundation, December 2004.
http://www.undl.org/unlsys/unl/UNLSpecs33.pdf

VAUQUOIS B. ET BOITET C. (1985) Automated translation at Grenoble University. *Computational Linguistics*, **11**/1, January-March 85, pp. 28—36.